I CLAIM

1. A method of forwarding protocol data units PDU's in a router with a forwarding hash table, comprising the steps of:

a) selecting a window size of *n* window bits and an offset of *o* offset bits;

b) generating a grouping table with sets of prefix lengths based on said window size and offset;

c) using said *n* window bits as a direct index into said grouping table to find an initial prefix length and provide an associated entry into said hash table; and

d) performing a lookup in said hash table based on said initial prefix length for matching said window bits with the bits of at said associated entry.

2. The method of claim 1, further comprising the steps of:

e) generating one of a hit pointer and a miss pointer in response to said lookup and loading said hit and miss pointers into a binary search tree; and

f) determining a next prefix length for a respective miss pointer and hit pointer; and

g) performing a further lookup in said hash table based on said next prefix length for matching said window bits with the bits of a further associated entry in said hash table.

3. The method of claim 2, further comprising repeating steps e), f) and g) until a longest matching prefix is obtained.

4. The method of claim 3, further comprising forwarding said PDU along a route identified by an IP address in said hash table corresponding to said longest matching prefix.

5. The method of claim 1, wherein step a) comprises selecting said offset such that the maximum number of prefix lengths per set is a minimum.

6. The method of claim 1, wherein said step a) comprises selecting said offset such that the average number of prefix lengths per set is minimized.

7. The method of claim 1, further comprising dynamically tuning the order of prefix lengths searched in said grouping table using statistical data collected at said router,

8. The method of claim 7, wherein said statistical data indicate the hits for each prefix length in each set.

9. The method of claim 7, wherein said lookup is dynamically tuned to process the prefix lengths in the order from the prefix lengths with a greater percentage of hits.

10. The method according to claim 1, wherein said hash table is updated when said window is updated.

11. The method of claim 1, wherein said window is a uni-dimensional window comprising a predetermined number of consecutive bits.

12. The method of claim 1, wherein said window is a multi-dimensional window comprising a predetermined number of groups of consecutive bits.

13. The method of claim 1, wherein the number of bits of said window is selected for enabling said grouping table to fit into a fast memory.

14. The method of claim 1, wherein the window size, *n*, is user-selectable.

15. The method of claim 1, wherein step a) comprises selecting said offset using said binary tree.

16. The method of claim 1, wherein said binary tree is made up of a root tree generated by said offset bits, partial trees within said window generated by said window bits, and sub-trees subtended by said partial trees.

17. The method of claim 16, wherein the number of prefix lengths searched for said window and said offset is a set union of the prefix lengths in said root tree, the prefix lengths of the sub-trees grouped by said window bits,

and the extended prefix lengths occurring within the window grouped by said window bits.

18. The method of claim 17, wherein said set union is obtained by iterating over all nodes of the tree within said window.

19. The method of claim 1, wherein said *n* and *o* are chosen periodically on a best effort basis.

20. The method of claim 1, wherein said *n* and *o* are chosen at router startup, and updated as a low priority background application.

21. A method of forwarding a protocol data unit (PDU) at a router with a forwarding hash table, comprising the steps of:

dividing the prefix lengths available in said hash table into groups; and

routing said PDU according to a lookup in said hash table based on the prefix lengths in a selected group of said groups.

22. The method of claim 21, wherein said step of dividing comprises selecting a set of *n* bits from the IP address of said PDU and arranging said selected group in the form of a grouping table with all prefix lengths available for said *n* bits.

23. The method of claim 22, wherein said step of routing comprises:

using said n bits for finding an initial prefix length in said grouping table, to determine an associated entry into said hash table; and

performing a lookup in said hash table based on said initial prefix length for matching said window bits with the bits of at said associated entry.

24. A memory for storing data for access by a routing program being executed on a router having a hash table, comprising:

a prefix length array for storing a grouping table comprising 2^n entries, each entry corresponding to a prefix length available for matching *n* bits of the IP address of a protocol data unit (PDU); and

a search area for storing a prefix length search tree constructed using said grouping table based on a lockup in said hash table, said lookup being performed for a prefix length in said grouping table, using said *n* bits in said IP address.

25. The memory of claim 24, wherein *n* is selected small enough so that said grouping table fits into a fast memory.

26. The memory of claim 24, wherein said prefix length search tree is constructed based on dynamic flow measurements to favour prefix lengths which are used by the majority of the PDUs at said router.